# CS 5012: Binary Trees Module 4.3 Exercise

**H. Diana McSpadden (hdm5s)**

**3/2/2021**

## Question 1:

Assume the implementation of a BinaryTree from slide 30 in *Introduction to Tree*s. Draw the binary tree that the following code produces (the code was written in *main*).

# create a bunch of nodes with integer values

theRoot = BinaryTree.Node(3) # root node with value 3

n1 = BinaryTree.Node(1)

n2 = BinaryTree.Node(2)

n4 = BinaryTree.Node(4)

n5 = BinaryTree.Node(5)

n6 = BinaryTree.Node(6)

# create a binary tree called 'myTree'

myTree = BinaryTree(theRoot) # create a tree 'myTree' with root = 3

# connect the tree

myTree.root.setLeft(n1)

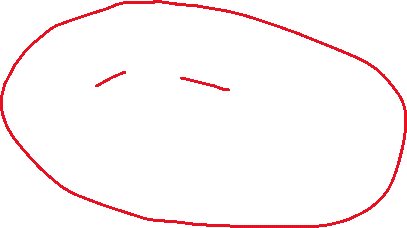
myTree.root.setRight(n4)

n1.setRight(n2)

n4.setRight(n6)



n4.setLeft(n5)



## Question 2:

Assume the implementation of a BinaryTree from slide 30 in *Introduction to Tree*s. Build a binary tree by providing the code in Python as would be typed in main (similar to the code in Q1), based on the following drawing.

**Answer**

# create the root node

theRoot = BinaryTree.Node(3)

# create the other nodes

n1 = BinaryTree.Node(1)

n2 = BinaryTree.Node(2)

n11 = BinaryTree.Node(11)

n5 = BinaryTree.Node(5)

n6 = BinaryTree.Node(6)

n9 = BinaryTree.Node(9)

# create the tree object

theTree = BinaryTree(theRoot)

# connect the tree

theTree.root.setLeft(n1)

theTree.root.setRight(n6)

n1.setLeft(n2)

n1.setRight(n11)

n6.setRight(n9)



n6.setLeft(n5)

## Question 3:

Write a tree-level *getHeight()* method that calculates the height (or depth) of the binary tree (e.g., it would return 3, if called on the tree in Q2).

Note: Think of a way to do this recursively, having each node calculate its height, as if it was the root of its subtree.

**Answer**

def getHeight(self):

if self.right and self.left:

rightHeight = getHeight(self.right)

leftHeight = getHeight(self.left)

if (leftHeight > rightHeight):

return (1 + leftHeight)

else:

return (1 + rightHeight)

elif node.left:

return (1 + getHeight(self.left))

elif node.right:

return (1 + getHeight(self.right))

else:

# height is 1 at a leaf, i.e. no children

return 1

root.getHeight()